An Architectural Model Framework to Improve Digital Ecosystems Interoperability

S. Shervin Ostadzadeh, Fereidoon Shams, and Kambiz Badie

Abstract

For the last two decades, software architecture has been adopted as one of the main viable solutions to address the ever-increasing demands in the design and development of complex software systems. Nevertheless, the rapidly growing utilization of communication networks and interconnections among software systems have introduced some critical challenges, which need to be handled in order to fully unleash the potential of these systems. In this respect, digital ecosystems, generally considered as a distributed adaptive open socio-technical system, have gained considerable attention, since their scale is incomparable to the traditional systems. The scale of socio-technical ecosystems makes drastic changes in various aspects of system development. As a result, it requires that we broaden our understanding of software architectures and the ways we structure them. In this paper, we investigate the lack of an architectural model framework for digital ecosystems interoperability, and propose an architectural model framework to improve digital ecosystems interoperability based on complex system theory.

Keywords

Framework • Architectural Model • Digital Ecosystems Interoperability • Software architecture • Software systems

Introduction

In today's digital era, the development of solid digital network infrastructures drastically affects the economic resources of communities as well as their lifestyle. In various domains, such as healthcare/health-science, energy, social networks, and logistics, future applications require infrastructures that are more agile than those functional at the moment. Digital ecosystems have emerged with the goal of capturing the notion of such agile and adaptive infrastructures. For this purpose, digital ecosystem technologies enclose the entire spectrum of Internet related technologies. This ranges from the hyperlinked web towards pervasive internet applications, and from peer-to-peer systems to Grid middleware. It also includes Cloud services, agent technologies, sensor networks, and cyber-physical systems. Thus, digital ecosystem has become one of the main topics for business process digitalization.

As systems grow larger and more complex to become digital ecosystems, new requirements for software architectures emerge. The software architecture of a program or computing system is the structure(s) of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among

S.S. Ostadzadeh (✉)
Computer Engineering Department, Science and Research Branch, Islamic Azad University, Tehran, Iran
e-mail: ostadzadeh@srbiau.ac.ir

F. Shams
Computer Engineering Department, Shahid Beheshti University, Tehran, Iran
e-mail: f_shams@sbu.ac.ir

K. Badie
IT Department, Iran Telecommunication Research Center, Tehran, Iran
e-mail: k_badie@itrc.ac.ir

K. Elleithy and T. Sobh (eds.), *New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering*, Lecture Notes in Electrical Engineering 312,
DOI 10.1007/978-3-319-06764-3_65,

them [1]. Based on this definition, it is inferred that software architecture characterizes the structure of a system. In general, architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution [2].

According to the ISO 15704 standard [3], an architecture represents a description of the basic arrangement and connectivity of parts of a system (either a physical or a conceptual object or entity), which is expected to create a comprehensive overview of the entire system when put together [4]. It should be noted that handling this large amount of information is quite challenging and needs a well-developed framework. The problem is even intensified in the case of digital ecosystems, due to their scale and socio-technical characteristics. So far, various Information Systems Architecture (ISA) frameworks have appeared in literature: Zachman framework [5], FEAF [6], TEAF [7], ToGAF [8], and C4ISR [9] to name a few. Nevertheless, these frameworks fail to provide all the required support for digital ecosystems. Consequently, the inability of current ISA frameworks to meet these requirements necessitates a breakthrough research in the development of a socio-technical ecosystems architectural framework [10].

In this paper, we present an architectural model framework in digital ecosystems interoperability based on complex system theory. The proposed framework is assumed to be capable of addressing the requirements of such systems.

The rest of the paper is organized as follows. In section "Helpful Hints", we present the required background and the problem definition. Next, we present an interoperability model overview in section "Interoperability Models". The digital ecosystems interoperability model based on complex system theory is discussed in section "Digital Ecosystems Interoperability". Finally, section "Conclusions" summarizes the contributions and sets the direction for the future work.

Helpful Hints

A digital ecosystem is a distributed adaptive open socio-technical system with properties of self-organization, scalability and sustainability inspired from natural ecosystems [11]. Digital ecosystems inherit concepts of open, loosely coupled, demand-driven, domain clustered, agent-based self-organized collaborative environments where species/agents form a temporary agreement for a specific purpose or goal. In such environments, the agents take precautionary measures and react for their own advantage. The adoption of ecological system concepts is the central characteristic of all digital ecosystems. This is achieved by bonding via collective intelligence and further collaboration instead of uncontrolled competition as well as ICT-based catalyst effects in various domains, to produce improved connected communities and solutions.

Ultra-Large-Scale Digital Ecosystems

In biology, an ecosystem is a community of plants, animals, and microorganisms that are linked by energy and nutrient flows and that interact with each other and with the physical environment. Rain forests, deserts, coral reefs, grasslands, and a rotting log are all examples of ecosystems. In 2006, SEI [10] published a report about some systems which were called as Ultra-Large-Scale (ULS) systems. ULS systems can be characterized as socio-technical ecosystems, whose elements are groups of people together with their computational and physical environments. These systems will go far beyond the size of current systems and system of systems by every measure, such as, the number of the lines of code; the number of people employing the system for different purposes; amount of data stored, accessed, manipulated, and refined; the number of connections and interdependencies among software components; and the number of hardware elements.

There are some characteristics of ULS systems that will be revealed because of their scale [10]: (1) decentralization; (2) inherently conflicting, unknowable, and diverse requirements; (3) continuous evolution and deployment; (4) heterogeneous, inconsistent, and changing elements; (5) erosion of the people/system boundary; (6) normal failures; (7) new paradigms for acquisition and policy. These characteristics undermine current, widely used, information systems framework and establish the basis for the technical challenges associated with ULS systems.

A complex systems is a system composed of interconnected parts that as a whole exhibit one or more properties (behavior among the possible properties) not obvious from the properties of the individual parts [12]. A system's complexity may be of one of two forms: disorganized complexity and organized complexity [13]. ULS digital ecosystems are examples of disorganized complexity, since disorganized complexity is a matter of a very large number of parts [14].

It has been observed that current approaches fail to fully define, develop, deploy, operate, acquire, and evolve ULS systems, as described in SEI report [10]. ULS systems are considered as cities or socio-technical ecosystems, while our current knowledge and practices are geared toward creating individual buildings or species. This inconsistency points out the research direction that is crucial for reaching a proper solution to develop ULS systems.

Research Context

The challenges that have to be addressed when developing a ULS digital ecosystem span three different areas: (1) Design and Evolution; (2) Orchestration and Control; and (3) Monitoring and Assessment [10]. The research work presented here addresses the design area related to "design and evolution".

Fundamental to the design and evolution of a digital ecosystem will be explicit attention to design across logical, spatial, physical, organizational, social, cognitive, economic, and other aspects of the system. Attention to design is also needed across various levels of abstraction involving hardware and software as well as procurers, acquirers, producers, integrators, trainers, and users. A key area of research in design is thus the need for design of all levels of a ULS digital ecosystem.

Why Interoperability?

Broadly speaking, interoperability refers to coexistence, autonomy, and federated environment, whereas integration conventionally refers to the concept of coordination, coherence, and uniformization [4]. ULS digital ecosystems go far beyond the size of current systems and system of systems by every measure, including, the number of the lines of code; the number of people using the system for different purposes; amount of data stored, accessed, manipulated, and refined; the number of connections and interdependencies among software components; and the number of hardware elements [10]. These are instances of "Loosely-Coupled" systems. This means that the components in such systems can interact and are connected by a communication network; they can exchange services while continuing locally their own logic of operation. "Tightly-Coupled" indicates that the components are interdependent and cannot be separated. This is the case of a fully integrated system. Thus, two integrated systems are inevitably interoperable; however, two interoperable systems are not necessarily integrated.

Problem Definition

The scale of complexity and uncertainty in the design of digital ecosystems is so immense that resists the treatments offered by traditional interoperability methods. According to SEI report [10], complexity is a new perspective: "architecture is not purely a technical plan for producing a single system or closely related family of systems, but a structuring of the design spaces that a complex design process at an industrial scale will explore over time". Breaking up an architecture into design spaces and striving for a set of coherent and effective design rules would seem to imply a significant degree of control of the overall design and production process. Nevertheless, the design spaces, design rules, and organizations will be continually adjusting and adapting to both internal and external forces, which makes it difficult to handle them all.

The criticality of the research is justified by the fact that handling the large volume of information available in ULS digital ecosystems is only feasible by utilizing a well-developed interoperability framework. A newly proposed framework is expected to broaden a traditional interoperability framework to include people and organizations; social, cognitive, and economic considerations; and design structures such as design rules and government policies.

This research work centers on the development of an architectural framework to improve the interoperability of digital ecosystems. We pose the question that given the issues with the design of all levels of ULS architectures, how can one organize and classify the types of information that must be created and used in order to improve the digital ecosystems interoperability?

Interoperability Models

Since the beginning of the last decade, most recent work on architecture development is focused on careful planning and improving an enterprise interoperability framework. Conventionally, such a framework is primarily concerned with establishing a mechanism to describe the concepts, the problem and the knowledge on enterprise interoperability in a more structured manner. This section will survey some recent interoperability models.

LISI Reference Model

The LISI [15] (Levels of Information Systems Interoperability) approach developed by C4ISR Architecture Working Group (AWG) during 1997, is a framework to provide the US Department of Defense (DoD) with a maturity model and a process for determining joint interoperability needs, assessing the ability of the information systems to meet those needs, and selecting pragmatic solutions and a transition path for achieving higher states of capability and interoperability.

A critical element of interoperability assurance is a clear prescription of the common suite of requisite capabilities that must be inherent to all information systems that desire to interoperate at a selected level of sophistication [4]. Each level's prescription of capabilities must cover all four enabling attributes of interoperability known as PAID, namely, Procedures, Applications, Infrastructure, and Data.

The LISI approach is focused on developing interoperability in US military sector. It is also used as a basis to elaborate other interoperability maturity models such as Organizational Maturity Model [16] and Enterprise Interoperability Maturity Model in ATHENA Integrated Project [17].

IDEAS Interoperability Framework

The IDEAS Interoperability Framework was developed by IDEAS project on the basis of ECMA/NIST Toaster Model, ISO 19101, ISO 19119 and was augmented through the quality attributes and intended to reflect the view that "Interoperability is achieved on multiple levels: inter-enterprise coordination, business process integration, semantic application integration, syntactical application integration and physical integration" [18].

In the business layer, all issues related to the organization and the management of an enterprise are addressed. The business model is the description of the commercial relationships between an enterprise and the way it offers products or services to the market. The knowledge layer is concerned with acquiring, structuring and representing the collective/personal knowledge of an enterprise. The ICT system layer is concerned with the ICT solutions that allow an enterprise to operate, make decisions and exchange information within and outside its boundaries. The semantic dimension cuts across the business, knowledge and ICT layers. Quality attributes are a supplementary dimension of the framework. The considered attributes are: (1) Security; (2) Scalability; (3) Portability; (4) Performance; (5) Availability; (6) Evolution.

ATHENA Interoperability Framework

The ATHENA Interoperability Framework (AIF) [17] is structured into three levels: (1) The 'Conceptual' level is used for identification of research requirements and integrates research results; (2) The 'Applicative' level is used for the transfer of knowledge regarding application of integration technologies; (3) The 'Technical' level is used for technology testing based on profiles and integrates prototypes.

The AIF and the IDEAS Interoperability Framework are considered complementary [4]. At each level of AIF, one can use the IDEAS interoperability framework to structure interoperability issues into three layers (business, knowledge and ICT) and a semantic dimension.

European Interoperability Framework

The European Interoperability Framework (EIF) [19] aims at supporting the European Union's strategy of providing user-centered e-Government services by defining as the overarching set of policies, standards and guidelines, which describe the way in which organizations have agreed, or should agree, to do business with each other.

This EIF is defined as the overarching set of policies, standards and guidelines, which describe the way in which organizations have agreed, or should agree, to do business with each other. EIF provides recommendations and defines generic standards with regard to organizational, semantic and technical aspects of interoperability.

Organizational Interoperability Maturity Model

Clark and Jones [16] proposed the Organizational Interoperability Maturity Model (OIM), which extends the LISI model into the more abstract layers of command and control support. OIM extends LISI to cover organizational interoperability. Five levels of organizational maturity, describing the ability to interoperate, are defined. These include: (1) Independent; (2) Ad-hoc; (3) Collaborative; (4) Combined; and (5) Unified.

Layers of Coalition Interoperability

Layers of Coalition Interoperability (LCI) [20] is a model to deal with possible measures of merit to be used with the various layers of semantic interoperability in coalition operations. In the LCI, interoperability is definitely not only limited to the technical domain, but also is dependent on organizational aspects.

LCI tries different aspects to implementing the relations between the interoperability levels. It also stresses the important role of a common unified language for the integration of the interoperability levels from technical up to organizational.

Other Relevant Interoperability Models

In the United Kingdom, the e-Government Unit7 (eGU), has based its technical guidance on the e-Government Interoperability Framework (e-GIF) [21]. e-GIF mandates sets of specifications and policies for any cross-agency collaboration and for e-government service delivery.

The E-health interoperability framework [22] developed by NEHTA (National E-Health Transition Authority) initiatives in Australia, brings together organizational,

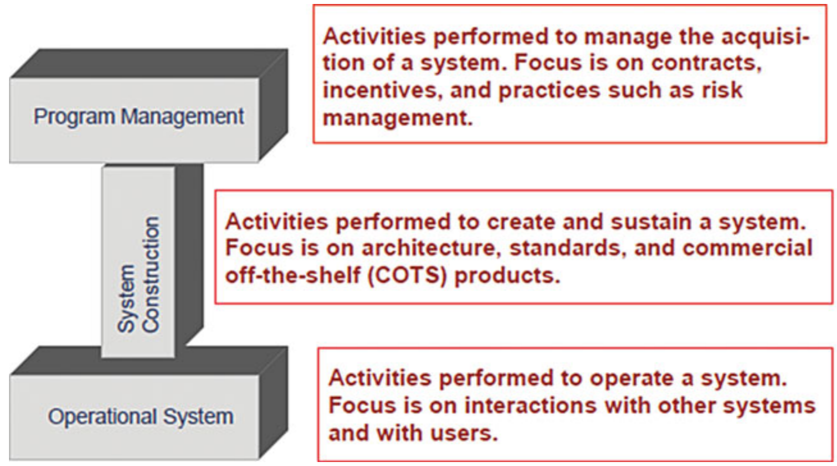

Fig. 1 System of systems interoperability [18]

information and technical aspects relating to the delivery of interoperability across health organizations.

The NATO C3 Interoperability Environment (NIE) [23] encompasses the standards, the products and the agreements adopted by the Alliance to ensure C3 interoperability. It serves as the basis for the development and evolution of C3 Systems.

The models previously discussed address a range of interoperability issues from technical to coalition organizational. SEI has developed the System of Systems Interoperability (SOSI) [24], which addresses technical interoperability (also covered by LISI, LCI, and NATO C3) and operational interoperability (also covered by OIM and LCI). However, SOSI goes a step further to address programmatic concerns between organizations building and maintaining interoperable systems. SOSI introduces three types of interoperability (see Fig. 1): (1) Programmatic, interoperability between different program offices; (2) Constructive, interoperability between the organizations that are responsible for the construction (and maintenance) of a system; (3) Operational, interoperability between the systems.

Digital Ecosystems Interoperability

As introduced in previous section, the SOSI can be considered as a significant initiative for digital ecosystems interoperability. However, as mentioned in SEI report [10], people will not just be users of a digital ecosystem; rather, they will be part of its overall behavior. In addition, the boundary between the system and user/developer roles will blur. Just as people who maintain and modify a city, may also reside in the city, in a digital ecosystem, a person may act in the role of a traditional user, or in a supporting role as a maintainer of the system health, or as a change agent adding and repairing the functions of the system.

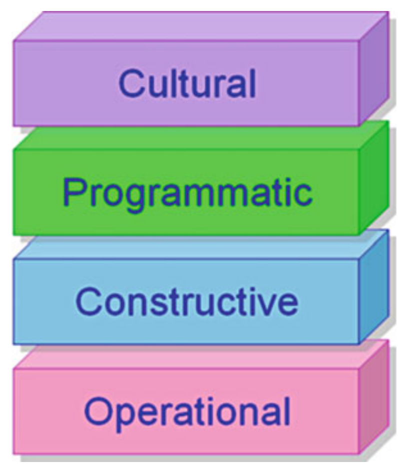

Fig. 2 Digital ecosystems interoperability model

Socio: The New Concern

Assuming that people are part of a digital ecosystem signifies that a new perspective has to be taken into account: culture. Figure 2 depicts an extension to the SOSI model in order to achieve socio-technical digital ecosystems characteristics.

The four layers of digital ecosystems interoperability model corresponds to the four layers of complex system theory model. In complex system theory, we can divide a system into four layers: (1) vital; (2) psyche; (3) social; (4) cultural [14].

The digital ecosystems interoperability model address a range of interoperability issues from operational to cultural. In order to achieve socio-technical interoperation among systems, a set of cultural, management, constructive, and operational activities have to be implemented in a consistent manner. These activities require to support adding new and upgraded systems to a growing interoperability web.

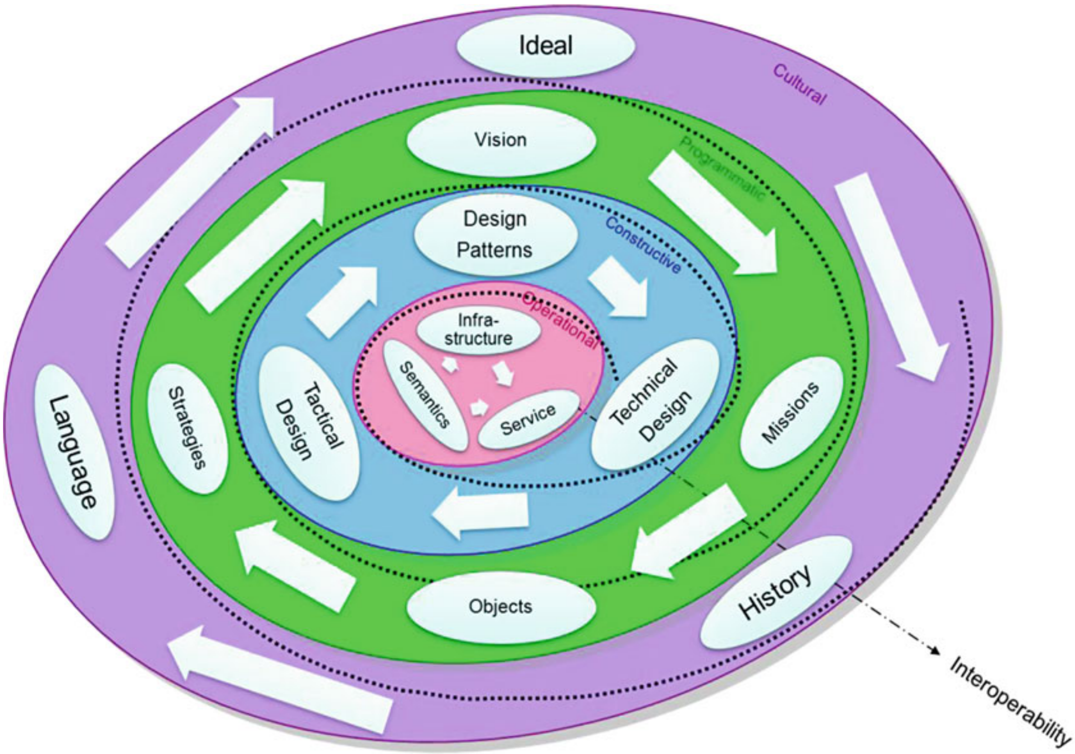

Fig. 3 Digital ecosystems interoperability model details [14]

- Operational issues define the activities within the executing system and between the executing system and its environment, including the interoperation with other systems.
- Constructive issues define the activities that develop or evolve system interoperability.
- Programmatic issues define the activities that manage the acquisition of system interoperability.
- Cultural issues define the activities that sustain the ULS system socio-technical characteristics.

As we mentioned earlier, the proposed model should be a spectrum of technologies and methods with software engineering, economics, human factors, cognitive psychology, sociology, systems engineering, and business policy. The Operational layer divided to three sub layers: (1) Infrastructure; (2) Service; (3) Semantics. The Constructive layer is divided to: (1) Tactical Design; (2) Technical Design; (3) Design Patterns.

The programmatic and cultural layers can also be divided to some sub layers. In the programmatic layer, we have four sub layers: (1) Vision; (2) Missions; (3) Objects; (4) Strategies. Finally, Cultural layer can be considered as combination of ideal, history, and language layers. Figure 3 depicts the details of digital ecosystems interoperability mode.

Digital Ecosystems Interoperability Framework

Zachman Framework (ZF) [5], originally proposed by John Zachman, is often referenced as a standard approach for expressing the basic elements of information system architecture, and is widely accepted as the main framework in ISA. Although some of today's successful ISA frameworks (including ZF) are used for enterprise systems architecture, the problem discussed in the previous section is inherently broader and deeper than current capabilities of ISA frameworks [25–27]. Figure 4 depicts our initiative proposed framework to improve interoperability based on complex system theory. In this work, we apply ZF as an initial start and try to enrich it by digital ecosystems interoperability model to support the special characteristics of socio-technical interoperability. The proposed framework should be a spectrum of technologies and methods with software engineering, economics, human factors, cognitive psychology, sociology, systems engineering, and business policy.

The proposed framework uses three basic dimensions: (1) The abstract dimension is based on six general questions required to understand interoperability; (2) The perspective dimension is based on interoperability concerns in an enterprise; (3) The final dimension is based on interoperability barriers in a socio-technical ecosystems.

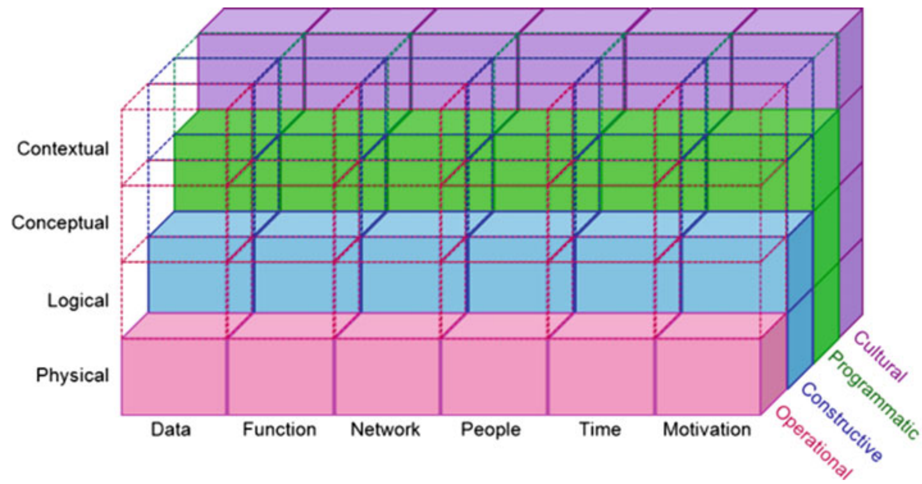

Fig. 4 Digital ecosystems interoperability framework. *Blank cells* are not supposed to be modeled

The interoperability abstracts define the contents of interoperations:

- Data (What?): The interoperability of data handles information finding and sharing from heterogeneous data sources. These data sources possibly exist within different machines running different operating and data management systems.
- Function (How?): The interoperability of function takes care of identifying, composing and making various application functions work together.
- Network (Where?): The study of interconnecting the internal networks of companies is essential in a networked enterprise. This facilitates the creation of a common network for the whole enterprise. This type of interoperability focuses on the geometry or connectivity of the system's physical nodes.
- People (Who?): It focuses on the people and the manuals and the operating instructions or models they use to interoperate their tasks/duties.
- Time (When?): It is concerned with the life cycles, the timing and the schedules used to interoperate activities.
- Motivation (Why?): It focuses on goals, plans and rules that prescribe policies and ends which guide the enterprise interoperability.

The interoperability perspectives define various concerns of interoperation:

- Contextual: It describes the artifacts that provide the boundaries for the interoperability.
- Conceptual: It focuses on the artifacts that conceptually define the interoperability from the enterprise owners' perspective.
- Logical: It describes the artifacts that design the way interoperability will be realized systematically, quite independently of any technologies.
- Physical: It focuses on the artifacts that define the interoperability implementation based on the general technological constraints being employed.

The interoperability barriers address a range of interoperability issues from operational to cultural. Together, the abstract, perspective and barrier dimensions constitute the digital ecosystems interoperability framework. The two dimensional matrix (abstract × perspective) defines the contents of interoperations that take place in various levels of system perspectives. The third dimension enables to capture and to structure the type of interoperation.

Conclusions

Achieving digital ecosystems interoperability involves changes to the way we define life, including acquisition practices and guidance, technologies, engineering and management practices, operational doctrines for both the usage and those who support the systems. Realizing this vision requires that we begin to define approaches and models in more concrete terms.

In this paper, an architectural framework to improve digital ecosystems interoperability was proposed. The framework presents a classification schema for descriptive representation of digital ecosystems and allows software architects to model various aspects of socio-technical interoperability. The goal is that the framework be used to complement a full-structural model within the socio-technical interoperability.

In the future work, one is expected to propose a methodology to help architectures model the framework cells.

References

1. L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 2nd Edition, SEI Series in Software Architecture, Addison-Wesley Professional, 2004.
2. Institute of Electrical and Electronics Engineers Standards board, *Recommended Practice for Architectural Description of Software-Intensive Systems*, IEEE-Std-1471-2000, 2000.
3. ISO 15704, *Industrial Automation Systems—Requirements for Enterprise-reference Architectures and Methodologies*, 2000.
4. D. Chen et al, "Architectures for enterprise integration and interoperability: Past, present and future," *Comput Industry (Ind)*, 2008.
5. J.A. Zachman, "A Framework for Information Systems Architecture," *IBM Systems Journal*, Vol. 26, No. 3, 1987, pp. 276–292. (Reprinted in 1999: Vol. 38, No. 2-3, 1999.)
6. Chief Information Officers (CIO) Council, *Federal Enterprise Architecture Framework*, Version 1.1, 1999.
7. Department of the Treasury, *Treasury Enterprise Architecture Framework*, Version 1, 2000.
8. Open Group, *The Open Group Architecture Framework (TOGAF)*, Version 9.0, USA, 2009.
9. C4ISR Architecture Working Group (AWG), *C4ISR Architecture Framework*, Version 2.0, USA. Department of Defense (DoD), 1997.
10. Software Engineering Institute (SEI), *Ultra-Large-Scale Systems: Software Challenge of the Future*, Technical Report. Carnegie Mellon University, Pittsburgh, PA, USA, 2006.
11. H. Zhuge and X. Shi, "Toward the Eco-grid: A Harmoniously Evolved Interconnection Environment," *Communications of the ACM*, 47(9)(2004)78-83.
12. C. Joslyn and L. Rocha, "Towards semiotic agent-based models of socio-technical organizations," *Proceedings of AI, Simulation and Planning in High Autonomy Systems (AIS 2000)* Conference, Tucson, Arizona, USA, 2000, pp. 70-79.
13. W. Weaver, "Science and Complexity," *American Scientist* 36: 536, 1948. (Retrieved on 2007–11–21.)
14. S.S. Ostadzadeh, B. Rezaei Rad, and F. Shams, "An Interoperability Architectural Model based On Complex System Theory for the Ultra-Large-Scale Systems," *Proceedings of Software Engineering and Application (SEA' 2011)*, IASTED, 2011.
15. C4ISR Architecture Working Group (AWG), *Levels of Information Systems Interoperability (LISI)*, 1998.
16. T. Clark and R. Jones, *Organizational Interoperability Maturity Model for C2*, Department of Defense, Canberra, Australia.
17. ATHENA, *Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications*, FP6-2002-IST1, Integrated Project, 2003.
18. IDEAS, *IDEAS: Interoperability Development for Enterprise Application and Software Roadmaps*, Annex 1—DoW, 2002.
19. EIF, *European Interoperability Framework for PAN-European eGovernment Services*, IDA Working Document, Version 4.2, 2004.
20. A. Tolk, "Beyond Technical Interoperability: Introducing a Reference Model for Measures of Merit for Coalition Interoperability," *Proc. 8th ICCRTS*, Washington ,USA, 2003.
21. eGovernment Unit, *eGovernment Interoperability Framework (eGIF)*, version 6.1, 2005.
22. NEHTA, *Towards an Interoperability Framework*, Version 1.8, 2005.
23. NC3A, *NATO C3 Technical Architecture Reference Model for Interoperability*, NATO Consultation, Command, and Control Agency, 2003.
24. E. Morris et al, *System of Systems Interoperability*, Technical Report. Carnegie Mellon University, Pittsburgh, PA, USA, 2004.
25. S. Blanchette et al, *U.S. Army Workshop on Exploring Enterprise, System of Systems, System, and Software Architectures*, Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, PA, USA, 2009.
26. P.J. Boxer and S. Garcia, "Enterprise Architecture for Complex System-of-Systems Contexts," *3rd Annual IEEE International Systems Conference*, Vancouver, Canada, 2009.
27. S.S. Ostadzadeh, F. Shams, and S.A. Ostadzadeh, "A Method for Consistent Modeling of Zachman Framework," *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, Springer, 2007, pp. 375-380.